Real-time Symbol Extraction from Grey-level Images

Massen R. Simnacher M. Rösch J. Herre E. Wuhrer H.W.

Transfer Centre Constance for Image Processing
Polytechnic of Constance
Reichenaustr. 81c, D-7750 Constance (FRG)

Abstract

A VME-bus image pipeline processor for extracting vectorized contours from grey-level images in real-time is presented. This 3 Giga operation per second processor uses large kernel convolvers and new non-linear neighbourhood processing algorithms to compute true 1-pixel wide and noise-free contours without thresholding even from grey-level images with quite varying edge sharpness. The local edge orientation is used as an additional cue to compute a list of vectors describing the closed and open contours in real-time and to dump a CAD-like symbolic image description into a symbol memory at pixel clock rate.

Iconic versus Symbolic Image Processing in Industry

Image processing for industrial inspection and machine vision demands for huge amounts of data to be processed in a very short time. Processing a grey-level image with 8 Bit intensity and 512 by 512 pixels spatial resolution in real-time, i.e. at frame rate easily requires computing powers of several Giga operation per second (GOPS). The traditional software based image processing methods working on pixel-wise digitized and stored images are becoming obsolete with the increasing resolution of matrix CCD-cameras like the 1300 by 100 pixel Megaplus Camera from Kodak and the use of laser scanners with 10 000 to 50 000 pixels resolution along a scan line. On the other hand, we are all aware that the human brain does not work pixel-wise, even if the retina is organized as a discrete grid of photosensitive pixels. We are not perceiving pixels, but image symbols or primitives like lines, edges, contours, corners, holes, textures and colour patches. These are far more condensed data structures then the highly redundant pixel or iconic image representations.

Research in symbolic image processing is under way for some time without however having produced much of real-life applications. The reason is not the lack of symbolic image processing methods, but the lack of symbols itself. Up to now, the extraction of even the most simple symbols like edges swallows enormous amounts of CPU time on a general purpose computer. Symbolic image processing simply is too slow and needs too much of computer power in order to be an industrial tool.

We report in this paper on the design of a pipelined hardware preprocessor for extracting vectorized contours from grey-level images at frame rate. These preprocessors use state of the art I.C.'s and achieve an equivalent computing power of several Giga OPS on a few VME-Bus boards. They allow software times for symbolic extraction to be cut by a factor of 5000 and bring symbolic image processing methods within the reach of industrial applications.

A pipelined image preprocessor

The methods for extracting even the most primitive symbols require extensive 2-D operations on a pixel based image data flow:

1. extracting thinned contours requires 2-D convolution with kernels from 3 by 3 up to 7 by 7 neighbours followed by additional 2-D non-linear neighbourhood processing.
2. vectorizing contours into linked lists requires 3 by 3 neighbourhood operations on contour pixel intensity and pixel orientation data.
3. matching reference shapes against unknown images requires very large kernel correlators and is probably best done simultanously on several levels of a pyramide image structure.
4. extracting texture features requires the processing of large windows in order to extract stable statistical parameters.

Even large arrays of transputers, image pipeline processors, geometric parallel processors and other non-von Neumann parallel computer architectures are not capable to achieve the Giga OPS goal required for real-time symbol extraction, at least not at reasonable cost and within reasonable space.

After a disappointing year of research in systolic arrays (1,2,3) and transputers, we found a synchronous pipeline of hardwired, but programmable 2-D neighbourhood processors to be the most successful way for achieving the Giga OPS goal with state of the art components at reasonable cost and confinable to a few boards of an industrial VME-Bus based image processing system (Fig.1).

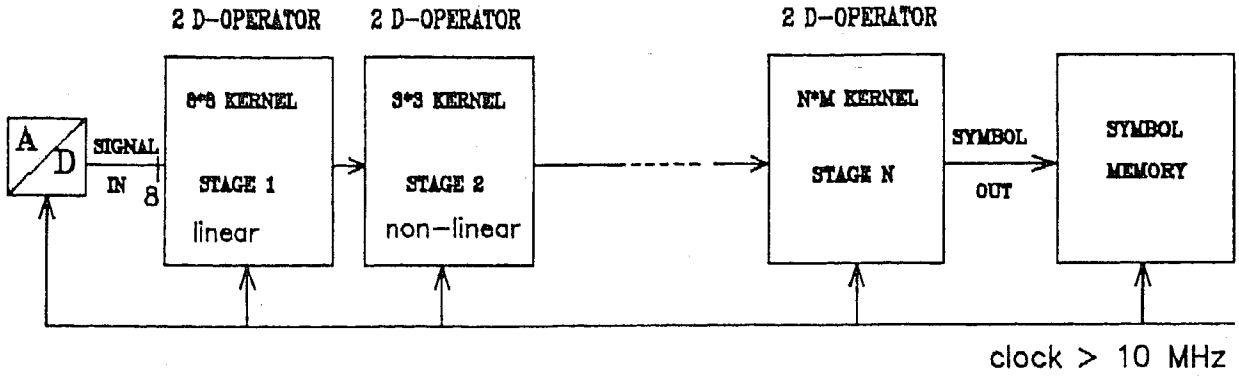

Fig.1 Real-time symbol extraction with a pipeline of 2-D processors

Real-time extraction of thinned contours

The lowest level of symbolic description for a typical grey-level image from industrial scene is a vectorized map of contours like those used in CAD-systems.

Fig.2 gives the diagram of our symbolic extraction pipeline we designed on 6 VME-Bus boards. The digitized pixel data flow coming from the video ADC is fed into two real-time convolvers. One is programmed as an 8 by 8 horizontal gradient filter, the other as a vertical gradient filter with the following set of coefficients:

$$Y = \begin{bmatrix} 0 & 1 & 2 & 0 & -2 & -1 & 0 \\ 1 & 2 & 4 & 0 & -4 & -2 & -1 \\ 1 & 3 & 5 & 0 & -5 & -3 & -1 \\ 2 & 4 & 7 & 0 & -7 & -4 & -2 \\ 1 & 3 & 5 & 0 & -5 & -3 & -1 \\ 1 & 2 & 4 & 0 & -4 & -2 & -1 \\ 0 & 1 & 2 & 0 & -2 & -1 & 0 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & -1 & -1 & -2 & -1 & -1 & 0 \\ -1 & -2 & -3 & -4 & -3 & -2 & -1 \\ -2 & -4 & -5 & -7 & -5 & -4 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 5 & 7 & 5 & 4 & 2 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 0 & 1 & 1 & 2 & 1 & 1 & 0 \end{bmatrix}$$

These boards use the TCD 1028 convolver chips from TRW and large FiFOs programmed as horizontal delay lines. Due to the limitations of the 10 MHz convolvers, both the video data and the filter coefficients have to be scaled to 4 Bits only. This coarse quantisation is however compensated by the large 64 pixel integrating neighbourhood of the 8 by 8 kernel.

The X and Y gradient images are fed into two look-up-tables which compute the gradient magnitude and the gradient orientation (Fig.2).

$$\text{GradMag} = \sqrt{x^2 + y^2}\,; \quad \text{GradAng} = \tan^{-1}(x/y) \qquad /1/$$

The gradient magnitude image is a multi-level intensity image with the magnitude of the gradient mapped into an 8 Bit value. The maximum intensities of such an image correspond to the points of inflection of the edges within the image. This is exactly the locus for placing a binary contour. It is completely wrong to extract the contour by simple thresholding. In a natural scene which contains both sharp and flat edges, the gradient magnitude varies strongly. Pure binarization will always produce noisy contours full of interruptions. This is often overlooked by commercial companies offering 2-D real-time convolver boards.

In order to find the contour irrespective of the amplitude of the gradient magnitude, we must search for the maximum in the gradient image (Fig.3). The maximum must be searched for along a line which is orthogonal to the local edge direction. The module "edge thinner" uses a bank of 8 comparators to compare the central pixel in 3 by 3 moving neighbourhood with those neighbours which are selected by the local edge orientation coming from the Grad-Angle look-up-table.

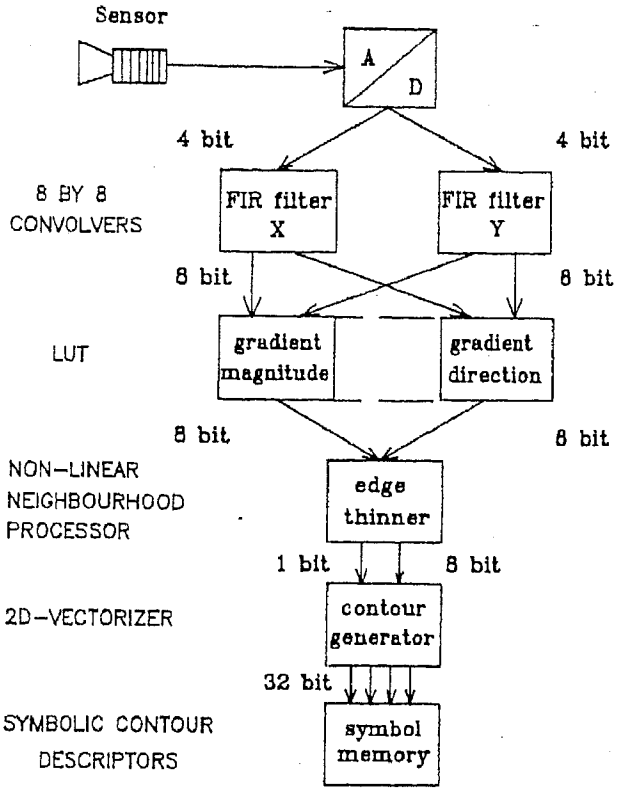

Fig.2 Real-time vectorized contour extractor

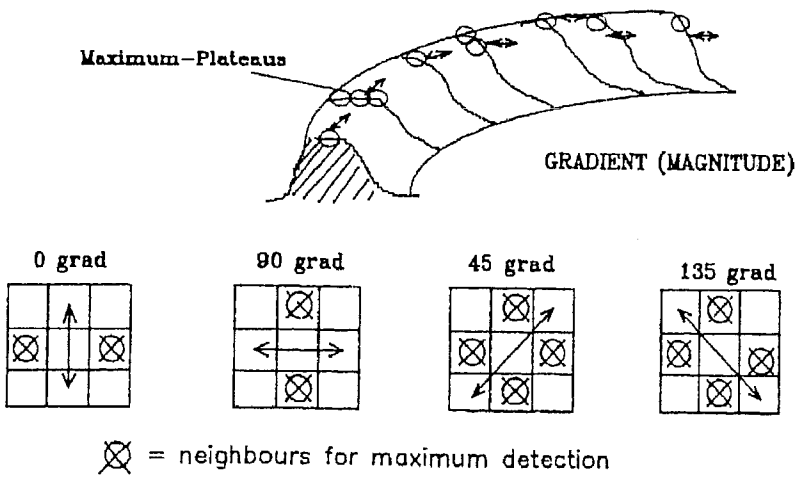

Fig. 3 Edge-Thinning Through Orientation-Controlled Maximum Detection (1)

Even these contours are not perfect. If the gradient magnitude image shows so-called gradient "plateaus" i.e. flat maxima of the gradient magnitude extending over several pixels, the contours extracted by the bank of comparators will show double, triple or even more parallel lines instead of being just a one pixel wide line contours. (Fig.4) We use a second 2-D neighbourhood processor working on this binarized contour. It performs an orientation controlled erosion and deletes on the fly all those contour pixels, which do not introduce holes in the continuous contour. Continuity is defined as an 8-connectivity; a line is considered as continuous if its pixels are horizontal, vertical or diagonal neighbours.

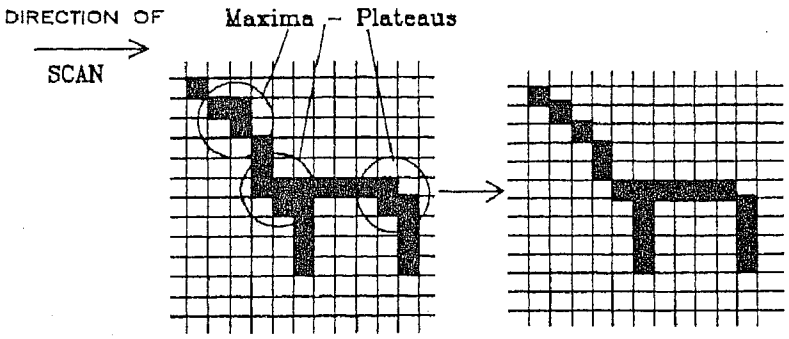

Fig.4 Recursive Orientation-Controlled Erosion

This real-time eroder works in direction of scanning, i.e. a multiple contour is eroded towards its right border, if scanning is done as usual from left to right. Fig. 5 shows the complete diagram of the 2-stage edge thinner module with the bank of orientation controlled comparators and the recursive orientation controlled eroder.

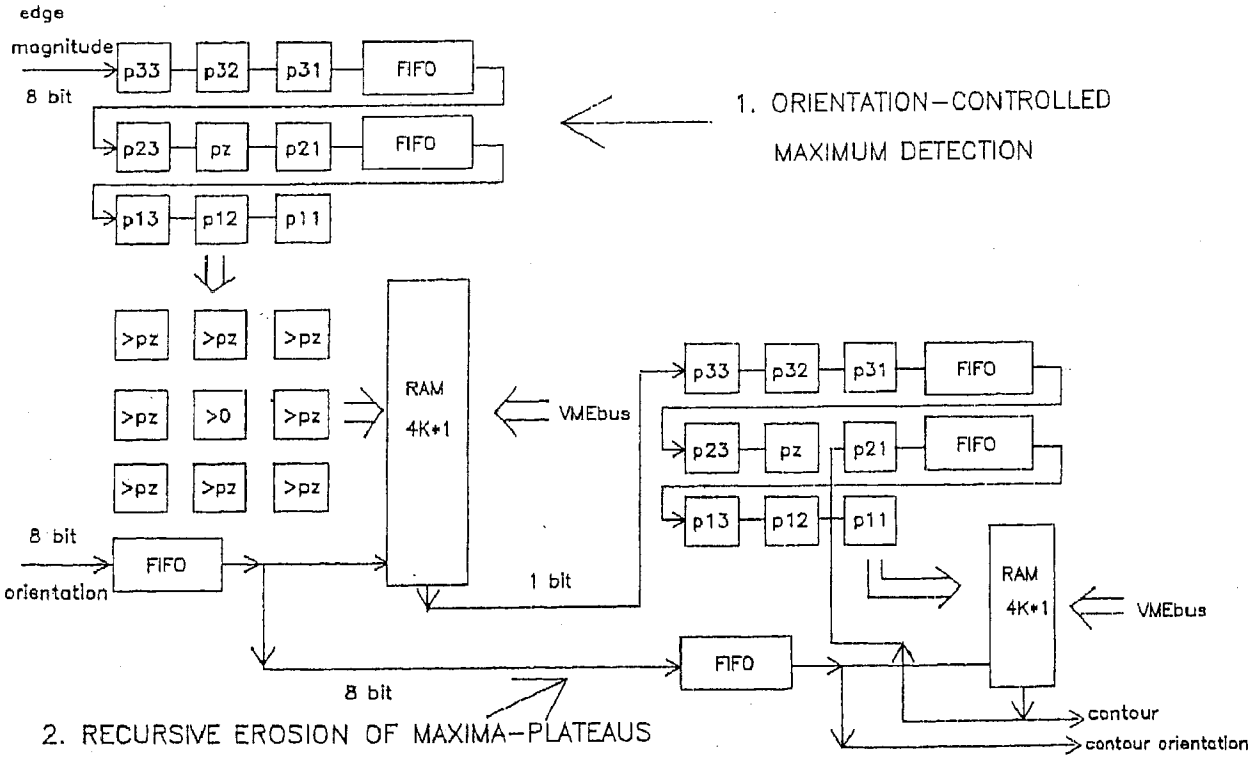

Fig. 5 Real-time gradient maximum detection and contour erosion

Real-time extraction of vectorized contour symbols

The output of the edge thinner module is still a pixel-based image, i.e. an iconic and not a symbolic image description. The most natural and simple symbols which can be extracted from such a line-drawing image are straight line segments, i.e. vectors. The next level could be higher-geometric symbols like corners, holes, T-and Y-branches etc.

We have developed a vector extraction module which is able to connect pixels lying on a straigth line into vectors in real-time (Fig. 8). The pixel image coming from the edge thinner module is again processed in a 2-D neighbourhood. If the local orientation between neighbouring pixels varies less than a given angle threshold, these neigbhours are connected into a vector. The start and end adresses of the vector are stored and a provisional contour index is attributed to it. Finding the beginnings, the endings and the branching of vectors is by no means a trivial task. It is realized with a PAL-based state automaton which will be described in more detail in a later publication.

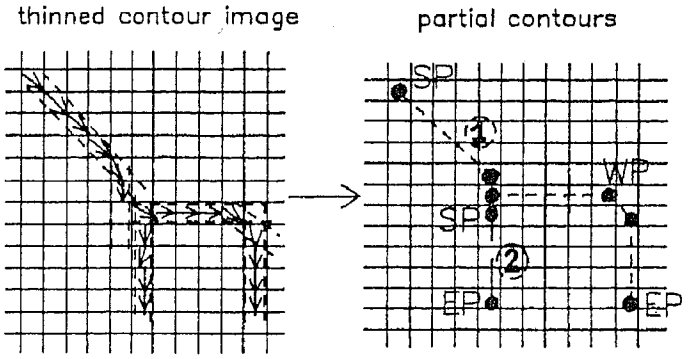

Fig. 8 Contour generator with real-time vectorization and labelling of thinned contours:

SP = start point of vector, EP = end point of vector, WP = "corner" point of contour

The sequences of vectors are starting and ending according to the direction of scan. If f.i. a circle is broken down into vectors, the vectorizer extracts two closed contours, one for the left half-circle and one for the right half-circle. These contours are stored as a sequence of "corner" adresses, a "corner" being those coordinates, where the local orientation changes more then the given threshold (Fig. 8).

A second stage of 2-D processing is needed to connect these two vectorized half-circles into a common closed contour by attributing linking pointers to the two half-circle contours. The final data structure which is dumped at pixel clock rate into the symbol memory is shown in Fig. 9 . It is almost equivalent to the data structures used for digital plotters and CAD stations.

Depending on how much linear segments are existing in the original pixel-based image, data reduction factors of 1000 to 10 000 are possible. Data reduction is however not the only goal. It is equally important that the industrial inspectio task can easily be preformed on the symbolic data structure. This is certainly the case for a contour vector structure. Comparing such a symbol list with a reference list, extracting hiher-order shape descriptors from these list etc. can easily be done by software using CAD-like algorithms.

The output of the edge thinner is an 8 Bit pixel image with 1 Bit of binary marked contour pixels and 7 Bits contour orientation data. It has to be emphasized that n thresholding at all has been used so far. The extracted thinned contours are of a hitherto unknown quality, even for natural scenes with grey-level edges of highly varying sharpness. Fig.6 gives us an example the thinned con contour from a poor contrast to those traditionally obtained by pure thresholding from a sobel-edge detector.

Our first pre processor was designed as a 4 board VME-Bu pipeline (2 convolver boards, 1 LUT-board and one edge thinner board). We have finished meanwhile a second gene ration board which concentrates all this circuitry on a single board. This "THINEDGE" board works with the 3 by 3 edge detector ship PSDP 16401 from Plessey and uses PALs for a modified neighbourhood processor. It can be retrofitted to any existing VME-bus image processing system by simple connection to the video ADC via an intermediate IC socket and a flat ribbon cable (Fig.7).

Fig.6 Thinned contour image of a poor contrast face

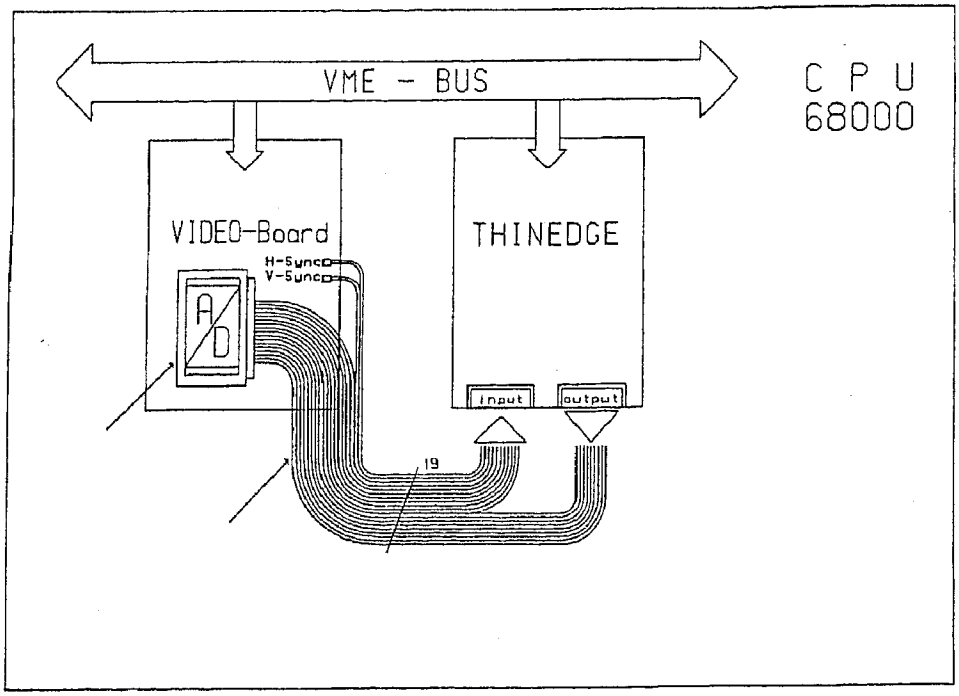

Fig.7 THINEDGE is a one-board real-time extractor of thinned contours from grey-level images. It can easily be retrofitted to any VME-bus system.

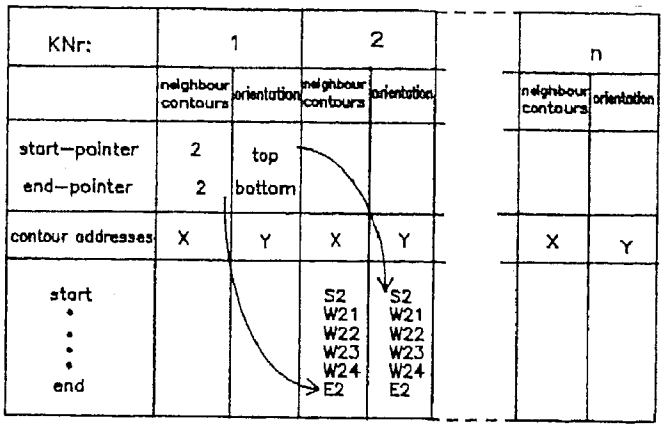

Fig. 9 symbolic data structure for linked vectorized contours

Conclusion

Extracting vectorized contour symbols from grey-level images without thresholding at 10 MHz pixel rate has been proofed feasible with a pipeline processor based on a few VME-bus boards and using state-of-the-art IC's. Obtaining a symbolic image description even with such simple primitives like vectorized contours in real-time brings speed-up factors of more than 10 000 compared to traditional purely software based symbol extraction. Powerful symbolic image processing and feature extraction algorithms are nor longer confined to pure mainframe-based image analysis but become an interesting tool even for time critical industrail applications.

References

1. Hillis, D.W., The Connection Machine, MIT Press 1986
2. Moore, W. et al. Systolic Arrays, Adam Hilger Publ. 1987
3. — · — Highly Parallel Signal Processing Architecturs Proc. SPIE, Vol 614 1986